METHOD OF PROCESSING A DIGITAL IMAGE

BACKGROUND

[0001]   A typical digital camera includes an array of photosensors, with each photosensor sensitive to only a single color of light. For example, each photosensor is sensitive to one of red, green and blue light.   During image acquisition, an image is focused on the photosensor array, and each photosensor measures or "samples" a single color of the image. If red-sensitive, green-sensitive and blue-sensitive photosensors can be located at each pixel, the photosensor array can acquire an image having "full color" at each pixel.

[0002]   The photosensor arrays of certain digital cameras have only a single photosensor at each pixel location. These cameras produce digital images that do not have full color information at each pixel. Since each photosensor is sensitive to only a single color, the photosensor array produces a digital image having only a single color sample at each pixel. For example, a digital camera produces a digital image having one of red, green and blue sampled information at each pixel. Information about the other two colors at each pixel is missing. This undersampled digital image is referred to as a "mosaic" image.

[0003]   A demosaicing algorithm may be used to transform an undersampled digital image into a digital image having full color information at each pixel value.   A typical demosaicing algorithm interpolates the missing pixel information from the sampled pixel values in the mosaic image.

[0004]   Edges and other abrupt photometric transitions present a particular problem to demosaicing.   A simple demosaicing algorithm such as bilinear interpolation fills in a missing pixel value by taking an average of sampled values from neighboring pixels. However, some of those neighbors might lie on opposite sides of an edge. Some of the neighbors lying on one side of the edge might belong to one object, while the other neighbors lying on the other side might belong to a different object. Consequently, the interpolated pixel information might not describe

either object. Since traditional bilinear interpolation does not account for edges, color information at edges in the demosaiced image can be distorted.

[0005]   More complex demosaicing algorithms try to account for edges. Still, even the more complex demosaicing algorithm can introduce artifacts into the demosaiced image. Zippering and fringing are typical artifacts at edges in the demosaiced image. These artifacts can degrade image quality.

[0006]   A typical digital camera can perform the demosaicing. This capability allows a typical digital camera to display and upload images having full color at each pixel. However, memory and processing power of a typical digital camera a re l imited. T he l imited m emory a nd processing power can constrain the complexity of the demosaicing algorithm and hamper the ability to account for edges and reduce artifacts at the edges during demosaicing.

[0007]   A demosaicing algorithm that is simple and fast, and that reduces edge blurring and the visibility of certain artifacts at edges, is desirable. Such a demosaicing algorithm is especially desirable for digital cameras.

SUMMARY

[0008]   According to one aspect of the present invention, processing of a mosaic digital image includes interpolating values of a first color at pixels where the first color was not sampled. The interpolation of the first color value at a given pixel includes determining likelihoods of the given pixel belonging to the same region as each of at least two other pixels having sampled values of the first color. The other pixels are in different directions relative to the given pixel. The interpolation further includes using the likelihoods and the sampled values of the other pixels to interpolate the first color at the given pixel.

[0009]   Other aspects and advantages of the present invention will become apparent from the following detailed description, taken in conjunction with the accompanying drawings, illustrating by way of example the principles of the present invention.

BRIEF DESCRIPTION OF THE DRAWINGS

[0010]   Figure 1 is an illustration of a photosensor arrangement for a typical digital camera, and corresponding pixels of a mosaic image.

[0011]   Figure 2 is an illustration of a method of performing image demosaicing in accordance with an embodiment of the present invention.

[0012]   Figure 3 is an illustration of an exemplary pixel neighborhood that can be used to interpolate missing green information at a given pixel in accordance with an embodiment of the present invention.

[0013]   Figure 4 is an illustration of a method of interpolating missing green information during d emosaicing i n a ccordance w ith a n e mbodiment o f t he p resent invention.

[0014]   Figure 5 is an illustration of a first method of interpolating missing red and blue information during demosaicing in accordance with an embodiment of the present invention.

[0015]   Figure 6 is an illustration of a second method of interpolating missing red and blue information during demosaicing in accordance with an embodiment of the present invention.

[0016]   Figure 7 is an illustration of a third method of interpolating missing red and blue information during demosaicing in accordance with an embodiment of the present invention.

[0017]   Figure 8 is an illustration of an apparatus for performing image demosaicing in accordance with an embodiment of the present invention.

[0018]   Figure 9 is an illustration of an apparatus for performing image demosaicing in accordance with an embodiment of the present invention.

DETAILED DESCRIPTION

[0019]   As s hown i n t he d rawings f or p urposes o f illustration, the present invention is embodied in the processing of a mosaic image to produce a digital image having full color information at each pixel.   Also for the purposes of illustration, the

demosaicing will be described in connection with a photosensor arrangement known as a Bayer color filter array (CFA).

[0020]   Reference is made to Figure 1, which illustrates a photosensor array 110 having photosensors 112 arranged in a Bayer CFA. The photosensors 112 are arranged in 2x2 cells 114. Each cell 114 consists of two photosensors sensitive to green (G) light only, one photosensor sensitive to red (R) light only, and one photosensor sensitive to blue (B) light only. The cells 114 are repeated (tiled) across the photosensor array 110.

[0021]   Figure 1 also illustrates a mosaic digital image 150. Each block 154 of the mosaic image 150 corresponds to a cell 114 of the photosensor array 110. Each pixel 152 of the mosaic image 150 is described by an n-bit word, and each n-bit word provides one of red, green and blue color sampled information. In each 2x2 block 154 of the mosaic image 150, green information is sampled at two pixels, red information is sampled at one pixel, and blue information is sampled at one pixel.

[0022]   Reference is made to Figure 2, which illustrates a method of processing a mosaic image into an output image having full color information at each pixel. Each pixel of the output image may be represented by a digital word including n-bits for red information, n-bits for green information, and n-bits for blue information.

[0023]   The sampled values of the mosaic image are copied into their corresponding positions in the output image (210). For example, a mosaic image with a Bayer pattern has pixel inputs $m(i,j)$, where i and j are indices to the rows and columns of the mosaic image. The sampled values of the mosaic image may be copied to the output image as follows:

$$I_R(i,j) = m(i,j) \text{ for all red samples (} i \text{ and } j \text{ even),}$$

$$I_G(i,j) = m(i,j) \text{ for all green samples (} i + j \text{ odd), and}$$

$$I_B(i,j) = m(i,j) \text{ for all blue samples (} i \text{ and } j \text{ odd),}$$

where $I_R(i,j)$ represents red information at pixel (i,j) of the output image, $I_G(i,j)$ represents green information at pixel (i,j) of the output image, and $I_B(i,j)$ represents blue information at pixel (i,j) of the output image.

[0024] The sampled values may be pre-processed (220). The pre-processing is application-specific. In a digital camera for example, pre-processing between image acquisition and demosaicing may include flare reduction, data linearization, color balance, and denoising. Denoising may be performed if the sampled pixel values (that is, the raw data) are noisy or expected to be noisy. Noisy raw data can cause artifacts and noise to appear in the demosaiced image.

[0025] Missing information in the green color plane is interpolated (230). For a given pixel that is missing green information, sampled values of nearby pixels are used to determine the likelihoods of the given pixel belonging to the same region as each of its neighbors having sampled green values. These neighbors are in different directions relative to the given pixel. For example, these neighbors might include a first pixel north of the given pixel, a second pixel south of the given pixel, a third pixel east of the given pixel, and a fourth pixel west of the given pixel.

[0026] A high likelihood suggests that the missing green value at the given pixel is likely to be close to the sampled value of the neighbor (e.g., an edge between the given pixel and the neighbor is not likely). Therefore, the sampled green value of the neighbor can be used with confidence in the interpolation of the green value at the given pixel. Thus a neighbor having a high likelihood is given a high weight in the interpolation process. Conversely, a low likelihood suggests that the given pixel is likely to be part of a different object than the neighbor (e.g., an edge between the given pixel and the neighbor is likely). Thus a neighbor having a low likelihood is given a low weight in the interpolation process.

[0027] The interpolation also includes computing correction terms for each direction. A correction term for a direction may be computed as a difference between the sampled value at the given pixel and a sampled value of the same color at a nearby pixel in that direction.

[0028] The interpolation further includes using the likelihoods, the sampled green values of the other pixels, and the correction terms to interpolate the green value at the given pixel. For example, the likelihoods are used to assign weights to the sampled values and the correction terms, and the green value at the given pixel

is interpolated as a weighted average of the sampled green values and the correction terms. The interpolated green value is added to the output image.

[0029] The use of the weighted correction terms may regarded as a high frequency correction to the interpolated green value. The same likelihood and weight may be used for the same direction. That is, the likelihood and weight computed for a measured green value may also be used for the correction term in the same direction. The meaning of "same direction" will become clear from examples below. In the alternative, likelihoods and weights may be computed separately for the correction terms.

[0030] Missing information in the other color planes is interpolated and added to the output image (240). Missing information in each of the red and blue color planes may be interpolated by using weighted averaging in the manner similar to that used to interpolate the missing information in the green color plane. Other methods of interpolating the missing information in each of the red and blue color planes include bilinear interpolation, affine interpolation, and a canonical correlation method, all of which will be described below. At the completion of step 240, each pixel $I(i,j)$ in the output image has red information $I_R(i,j)$, green information $I_G(i,j)$, and blue information $I_B(i,j)$.

[0031] Post processing may be performed on the output (demosaiced) image (250). The post-processing is application-specific. In a digital camera for example, the post-processing may include color transformation, tone reproduction, sharpening, and compression.

[0032] Reference is now made to Figures 3, which shows a few pixels of the output image after the sampled values are copied to it, but before any of the missing values are interpolated. The letters R, G and B denote the sampled colors at the pixels. Arrows denote possible directions. The labels N, S, E and W identify these possible directions as north, south, east and west (these directions are figurative, not literal). The north and south directions are orthogonal to the east and west directions.

[0033] Additional reference is made to Figure 4, which illustrates a method of interpolating the missing information in the green color plane. The method may be performed for each pixel that is missing green information in the mosaic image.

[0034] By way of example, the method will be described for the pixel that is circled in Figure 3. The given pixel in Figure 3 has a sampled red value, but is missing green and blue information. Sampled green values are at pixels to the immediate north, south, east and west of the given pixel.

[0035] Sampled green values of the neighboring pixels are used to determine the likelihoods of the given pixel belonging to the same region as each of those neighboring pixels (410). Considering the example of Figure 3, a first likelihood of the given pixel belonging to the same region as the north neighbor is determined, a second likelihood of the given pixel belonging to the same region as the south neighbor is determined, a third likelihood of the given pixel belonging to the same region as the east neighbor is determined, and a fourth likelihood of the given pixel belonging to the same region as the west neighbor is determined.

[0036] Each likelihood may be based on differences of the sampled green values. For example, the first likelihood may be computed from a difference between the sampled green values of the north and south neighbors; the second likelihood may be computed from a difference between the sampled green values of the north and south neighbors; the third likelihood may be computed from a difference between the sampled green values of the east and west neighbors; and the fourth likelihood may be computed from a difference between the sampled green values of the east and west neighbors.

[0037] Each likelihood may be also based on differences between the sampled red value of the given pixel and the sampled red value closest to the given pixel in the corresponding direction. For example, the first likelihood term for the north neighbor also may be computed from a difference between the sampled red value of the given pixel and the sampled red value north of the given pixel. The second likelihood term for the south neighbor also may be computed from a difference between the sampled red value of the given pixel and the sampled red value south of the given pixel. The third likelihood term for the east neighbor also may be computed

-7-

from a difference between the sampled red value of the given pixel and the sampled red value east of the given pixel. The fourth likelihood term for the west neighbor also may be computed from a difference between the sampled red value of the given pixel and the sampled red value west of the given pixel.

[0038]   The likelihood of a given pixel belonging to the same region as a neighboring pixel may be determined by applying a similarity function to the sum of absolute differences corresponding to that direction. A large absolute difference in sampled values over a short distance (that is, an abrupt transition) is assumed to result from an edge. Thus a large sum of absolute differences in a given direction implies that the given pixel lies on one side of an edge (and belongs to one object), and the neighbor at that direction lies on the other side of the edge (and belongs to a different object). Therefore, a low likelihood exists that the green values of the given pixel and neighbor are similar. Exemplary similarity functions $\rho(x)$ include, but are not limited to, the following:

$$\rho(x) = \exp(\frac{-x^2}{\sigma^2})$$

$$\rho(x) = \frac{1}{|x| + 1}$$

$$\rho(x) = \min(\frac{T}{|x|}, 1)$$

$$\rho(x) = 1 / \max(|x|, 1)$$

where T and σ are scaling parameters that account for the expected noise level in the raw data. The input x corresponds to a sum of absolute differences.

[0039]   Correction terms are also computed (420). A first correction term for the north neighbor may be computed as a difference between the sampled red value of the given pixel and the sampled red value north of the given pixel. A second correction term for the south neighbor may be computed as a difference between the sampled red value of the given pixel and the sampled red value south of the given pixel. A third correction term for the east neighbor may be computed as a difference between the sampled red value of the given pixel and the sampled red value east of the given pixel. A fourth correction term for the west neighbor may be computed as a

difference between the sampled red value of the given pixel and the sampled red value west of the given pixel.

[0040] The likelihoods, the sampled pixel values, and the correction terms are used to interpolate the missing green value at the given pixel (430). For example, the missing green pixel value (I_G) at a given pixel (i, j) may be computed as follows:

$$I_G(i,j) = \frac{\sum_{(a,b)\in N_4}\{\lambda[m(i,j) - m(i+2a, j+2b)] + m(i+a, j+b)\}w_{a,b}(i,j)}{\sum_{(a,b)\in N_4} w_{a,b}(i,j)}$$

where N_4 is the neighborhood consisting of the four sampled green values. The index (a,b) indicates the direction and number of pixels relative to the given pixel. For example, the neighborhood N_4 may consist of {(-1,0), (1,0), (0,-1) and (0,1)}, which are the nearest pixels due west (-1,0), due east (1,0), due south (0,-1) and due north (0,1) of the given pixel. The above formula holds only for those pixels (i,j) that correspond to non-green positions in the mosaic image.

[0041] The bracketed term $[m(i,j) - m(i+2a, j+2b)]$ represents the correction term for sampled green value at the neighbor (a, b), and the term $m(i+a, j+b)$ represents the sampled green value at the neighbor (a, b).

[0042] The parameter λ is a tunable parameter that corrects the interpolated value for camera-specific items, such as image resolution, camera optics, and scene distance. If device resolution is small or edges are too abrupt, the parameter λ may have a value between 0.5 and 1. For mid-resolution cameras, the parameter λ may have a value of 0.5. For high resolution cameras, the parameter λ may have a value of less than 0.5.

[0043] The weights $w_{a,b}$ may be computed as

$$w_{a,b}(i,j) \overset{\Delta}{=} \rho\left[|m(i+2a, j+2b) - m(i,j)| + |m(i+a, j+b) - m(i-a, j-b)|\right]$$

The first absolute difference $\left|m(i+a,j+b)-m(i+a,j+b)\right|$ represents the differences in the green color plane, and the second absolute difference $\left|m(i+2a,j+2b)-m(i,j)\right|$ represents the difference in the red color plane.

[0044]   The method is not limited to a Bayer CFA. The method may be applied to other color filter arrays.

[0045]   The method is not limited to the N, S, E and W directions, or the four closest neighbors. Larger neighborhoods may be used, and pixels of different distances may be used. Weights may be reduced as distance increases.

[0046]   Various methods may be used to interpolate the missing information in the red and blue color planes. One such method is bilinear interpolation of corrected values. In bilinear interpolation, likelihoods are not computed, and all weights are the same. The interpolation may be based on a smaller neighborhood because correction terms can now be computed entirely from the green plane, which is now fully populated. Therefore, differences may be taken from adjacent green pixels, as opposed to differences taken from samples that are two pixels apart (which occurs during interpolation of the green plane). The following formula may be used to interpolate a missing red or blue color $I_c(i,j)$ at pixel (i,j), given the fully populated green plane I_G :

$$I_c(i,j) = \frac{1}{N_{G,c}^{(i,j)}} \sum_{(a,b) \in N_{G,c}^{(i,j)}} I_G(i,j) - I_G(i+a,j+b) + m(i+a,j+b)$$

where $c \in \{R,B\}$, and

$$N_{G,c}^{(i,j)} \begin{cases} \{(-1,0),(1,0)\}, & \begin{array}{l} c = R,\ i\ is\ odd,\ j\ is\ even,\ or \\ c = B,\ i\ is\ even,\ j\ is\ odd \end{array} \\ \{(0,-1),(0,1)\}, & \begin{array}{l} c = B,\ i\ is\ odd,\ j\ is\ even,\ or \\ c = R,\ i\ is\ even,\ j\ is\ odd \end{array} \\ \{(-1,-1),(-1,1)\},\{(1,-1),(1,1)\}, & \begin{array}{l} c = R,\ i\ is\ odd,\ j\ is\ odd,\ or \\ c = B,\ i\ is\ even,\ j\ is\ even \end{array} \end{cases}$$

[0047]   Figures 5-7 illustrate other methods of interpolating the missing information in the red and blue color planes. Reference is now made to Figure 5, which illustrates a method that is essentially the same as the method used to interpolate the missing green information. The nearest sampled red or blue value in

each d irection i s t wo p ixels f rom t he g iven p ixel. At step 510, the likelihoods are determined from the green plane only. In addition, the correction terms are determined from the green plane (step 520). At step 530, the following formula may be used to interpolate a missing color $I_c(i,j)$ at pixel (i,j):

$$I_c(i,j) = \frac{\sum_{(a,b)\in N_{G,c}^{(i,j)}} \left[I_G(i,j) - I_G(i+a,j+b) + m(i+a,j+b)\right] u_{a,b}(i,j)}{\sum_{(a,b)\in N_{G,c}^{(i,j)}} u_{a,b}(i,j)}$$

where $c \in \{R,B\}$, and

$$N_{G,c}^{(i,j)} \begin{cases} \{(-1,0),(1,0)\}, & \begin{array}{l} c = R, i \text{ is odd, } j \text{ is even, or} \\ c = B, i \text{ is even, } j \text{ is odd} \end{array} \\ \{(0,-1),(0,1)\}, & \begin{array}{l} c = B, i \text{ is odd, } j \text{ is even, or} \\ c = R, i \text{ is even, } j \text{ is odd} \end{array} \\ \{(-1,-1),(-1,1)\},\{(1,-1),(1,1)\}, & \begin{array}{l} c = R, i \text{ is odd, } j \text{ is odd, or} \\ c = B, i \text{ is even, } j \text{ is even} \end{array} \end{cases}$$

and

$$u_{a,b}(i,j) \overset{\Delta}{=} \rho\left[\left|I_G(i,j) - I_G(i+a,j+b)\right|\right].$$

[0048]   Reference is now made to Figure 6, which illustrates a canonical correlation method of interpolating the missing red and blue information. An initial estimate for the missing values is made (610). For instance, the initial estimate may be made by using standard bilinear demosaicing. These initial estimates are added to the output image. Thus, at the end of step 610, the output image includes sampled a nd i nterpolated v alues f or t he green plane, and sampled and estimated values for the red and blue planes.

[0049]   The output image is transformed into a luminance-chrominance color space (620). For example, the output image may be transformed into YIQ color space.

[0050]   The chrominance bands of the YIQ color space are strongly smoothed (630). The luminance band may be lightly smoothed; left unchanged, or even lightly sharpened (630). According to the principles achieved by canonical correlation analysis, natural images are expected to be smooth in chrominance

bands, whereas all sharpness that the human eye perceives is expected to stem from the luminance band only. In terms of frequencies, light smoothing would be less aggressive in reducing high-frequencies of an image, as compared to the strong smoothing.

[0051]   The output image is transformed back to its original color space (640), and reset to the sampled values and green interpolated values (650). Step 650 imposes a constraint on the interpolation.   After filtering in the YIQ domain, the sampled values in the green plane of the output image are likely to change from their original values. The previously calculated green plane is likely to have changed as well. Imposing the constraints or resetting the sampled and green interpolated values is performed to replace the output green plane by the previously calculated green plane, and to reset the original red and blue mosaic samples on their corresponding positions.

[0052]   Steps 620 to 650 may be repeated several times. The steps 620 to 650 may converge to an optimal result. The iterations can be stopped when the result is "close enough" to the optimum. For example, the iterations can be stopped when the improvement (Euclidean distance between the image before and after an iteration) becomes smaller than a threshold. In the alternative, a fixed number of iterations can be performed. The fixed number would typically bring the output close enough to the optimum.

[0053]   Provided that the smoothing is linear, a fixed number of iterations can be concatenated into a single linear operation to provide a set of twelve linear convolution kernels. A set of kernels can be convolved with the pixel values of each cell.

[0054]   Reference is now made to Figure 7, which illustrates affine interpolation of the missing red and blue information. The following formula may be used to interpolate a missing color $I_c(i,j)$ at pixel (i,j):

$$I_d(x,y) = t_{d,c}(x,y)I_c(x,y) + s_{d,c}(x,y)$$

where $d,c \in \{R,G,B\}$, $s_{d,c}$ is a slowly varying function of a shift in color c, and $t_{d,c}$ is a slowly varying function that scales the intensity gradient.

-12-

[0055] The function $s_{d,c}$ is based on the assumption that object boundaries in a small spatial neighborhood are the same in all three color planes, or that the high frequency components (edges depicting object boundaries) in a small spatial neighborhood are similar in all three planes and equal with the high frequency component of the green plane.

[0056] The function $t_{d,c}$ is based on the assumption that an edge in a color plane can vary by a different amount than the same edge in another color plane.

[0057] The functions $s_{d,c}$ and $t_{d,c}$ can be solved for from the green plane. Consider a pixel (i_0, j_0) having missing red information. Red information $I_R(i_q, j_q)$ and green information $I_G(i_q, j_q)$ are known at neighboring pixels (i_q, j_q), $q = 1, ..., k$, where either i_0 or j_0 are odd, where i_q and j_q are even, and where $(i_q, j_q) \in N(i_0, j_0)$. The following set of equations is obtained.

$$\left\{ I_R(i_q, j_q) = t_{R,G}(i_0, j_0) I_G(i_q, j_q) + s_{R,G}(i_0, j_0) \right\}_{q=1}^{k}.$$

[0058] These inequalities can be rewritten in matrix form:

$$\begin{pmatrix} I_G(i_1, j_1) & 1 \\ I_G(i_2, j_2) & 1 \\ \vdots & \vdots \\ I_G(i_k, j_k) & 1 \end{pmatrix} \begin{pmatrix} t_{R,G}(i_0, j_0) \\ s_{R,G}(i_0, j_0) \end{pmatrix} = \begin{pmatrix} I_R(i_1, j_1) \\ I_R(i_2, j_2) \\ \vdots \\ I_R(i_k, j_k) \end{pmatrix}$$

[0059] A Mean Least Squares solution gives the following

$$\hat{t}_{R,G}(i_0, j_0) = \frac{E\{I_G I_R\}(i_0, j_0) - E\{I_G\}(i_0, j_0) E\{I_R\}(i_0, j_0)}{E\{I_G^2\}(i_0, j_0) - E^2\{I_G\}(i_0, j_0)} \quad \text{and}$$

$$\hat{s}_{R,G}(i_0, j_0) = E\{I_R\}(i_0, j_0) - \hat{t}_{R,G}(i_0, j_0) E\{I_G\}(i_0, j_0) E\{I_G\}(i_0, j_0)$$

where

$$E\{f\}(i_0, j_0) = \frac{1}{k} \sum_{(i_q, j_q)} f(i_q, j_q) \quad \text{for } (i_q, j_q) \in N(i_0, j_0) \text{ where } i_q \text{ and } j_q \text{ are even}.$$

[0060] Neighborhood size is based on assumptions about how quickly $t_{d,c}$ and $s_{d,c}$ change. A 5x5 neighborhood has been found to provide a reasonable tradeoff of accuracy versus stability. Increasing the neighborhood size can improve

the accuracy, but decrease the stability, since there is a greater chance that $t_{d,c}$ and $s_{d,c}$ will not remain constant.

[0061]   The operator E{} may be regarded as a simple averaging of neighborhood. The following equation, which can be used instead, produces a weighted average.

$$E\{f\}(i_0,j_0) = \frac{1}{k} \frac{\sum_{(i_q,j_q)} f(i_q,j_q) u_q(i_0,j_0)}{\sum_{(i_q,j_q)} u_q(i_0,j_0)}$$

where $u_q(i_0,j_0) = \rho\left(\left|I_G(i_0,j_0) - I_G(i_q,j_q)\right|\right)$.

[0062]   Reference is made to Figure 8, which illustrates an exemplary hardware implementation of the methods above. A digital camera 810 includes a photosensor array 812 such as a CCD. The photosensors of the array 812 may be arranged in a Bayer CFA. The array 812 produces mosaic digital images.

[0063]   The digital camera 810 also includes a digital signal processor 814 and memory 816 (e.g., EPROM). A program 818 encoded in the memory 816 instructs the digital signal processor 814 to transform mosaic images into digital images having full color information at each pixel in accordance with an embodiment of the present invention. The program 818 may include a lookup table for different values of the similarity function. Weights are computed on the fly, and applied to the pixels using an algorithm.

[0064]   Another implementation (not shown) may include an application-specific integrated circuit (ASIC). The ASIC includes a processor for transforming mosaic images into digital images having full color information at each pixel in accordance with an embodiment of the present invention. The processor may include a state machine, or it may include an embedded processor and memory, or it may have some other design.

[0065]   Reference is now made to Figure 9, which illustrates another hardware implementation. A digital imaging system 910 includes a general purpose processor 912 and memory 914 for instructing the processor 912 to process mosaic images in accordance with an embodiment of the present invention. The system may be, for example, a personal computer. Mosaic images may be supplied to the system

910 directly by a capture device (e.g., a scanner, a digital camera) or indirectly (via the Internet, accessed from remote or local storage, etc.).

[0066]   A machine for generating the program may be a personal computer such as the one shown in Figure 9.   Source code is generated, compiled into executable code, and distributed.

[0067]   The executable code may be distributed in variety of ways. For example, the executable code may be distributed to a digital camera by burning the code into an EPROM of the digital camera. The executable code can be distributed to a computer via a network transmission, removable storage medium (e.g., CD), etc.

[0068]   The present invention is not limited to the specific embodiments described above.   Instead, the present invention is construed according to the following claims.